IT & C

ISSN 2821 - 8469, ISSN – L 2821 - 8469, Volumul 2, Numărul 3, Septembrie 2023

Argumentele funcțiilor PHP - Transmiterea argumentelor prin referință

Nicolae Sfetcu

Sfetcu, Nicolae (2023), Argumentele funcțiilor PHP - Transmiterea argumentelor prin referință, *IT & C*, **2**:3, 53-57, DOI: 10.58679/IT35038, https://www.internetmobile.ro/argumentele-functiilor-php-transmiterea-argumentelor-prin-referinta/

Publicat online: 02.08.2023

Argumentele funcțiilor PHP - Transmiterea argumentelor prin referință

Nicolae Sfetcu[1]
nicolae@sfetcu.com

PHP function arguments - Passing arguments by reference

Abstract

In PHP, function arguments are the parameters that a function receives when called. These parameters are placeholders for the actual data that will be provided by the caller. PHP function arguments can be of different types and each type serves a specific purpose. PHP function arguments are a fundamental aspect of the language that allows developers to create dynamic and flexible functions.

Keywords: function arguments, PHP functions, passing arguments, reference, PHP

Rezumat

În PHP, argumentele funcției sunt parametrii pe care îi primește o funcție atunci când este apelată. Acești parametri sunt substituenți pentru datele reale care vor fi furnizate de apelant. Argumentele funcției PHP pot fi de diferite tipuri și fiecare tip servește unui scop specific. Argumentele funcției PHP sunt un aspect fundamental al limbajului care permite dezvoltatorilor să creeze funcții dinamice și flexibile.

Cuvinte cheie: argumentele funcțiilor, funcții PHP, transmiterea argumentelor, referință, PHP

IT & C, Volumul 2, Numărul 3, Septembrie 2023, pp. 53-57
ISSN 2821 - 8469, ISSN – L 2821 – 8469, DOI: 10.58679/IT35038
URL: https://www.internetmobile.ro/argumentele-functiilor-php-transmiterea-argumentelor-prin-referinta/

[1] Cercetător - Academia Română - Comitetul Român de Istoria și Filosofia Științei și Tehnicii (CRIFST), Divizia de Istoria Științei (DIS)

Introducere

PHP (Hypertext Preprocessor) este un limbaj de scripting folosit pe scară largă pe server, care permite dezvoltatorilor să construiască aplicații web dinamice și interactive. Una dintre caracteristicile esențiale ale PHP este capacitatea sa de a defini și utiliza funcții. Funcțiile sunt blocuri de cod reutilizabil care pot fi executate ori de câte ori este necesar și joacă un rol crucial în îmbunătățirea organizării codului, a menținabilității și a reutilizabilității. Argumentele funcției sunt intrări transmise unor funcții care le permit să proceseze diferite date și să îndeplinească sarcini specifice.

În PHP, argumentele funcției sunt parametrii pe care îi primește o funcție atunci când este apelată. Acești parametri sunt substituenți pentru datele reale care vor fi furnizate de apelant. Când este definită o funcție, dezvoltatorul specifică numărul necesar de argumente și tipurile acestora în semnătura funcției. Când funcția este invocată, apelantul furnizează valorile reale pentru aceste argumente.

Argumentele funcțiilor

Argumentele funcției PHP pot fi de diferite tipuri și fiecare tip servește unui scop specific:

- **Argumente necesare**: Aceștia sunt parametrii pe care trebuie să îi primească o funcție pentru ca aceasta să funcționeze corect. Dacă lipsește vreun argument necesar atunci când funcția este apelată, PHP va genera o eroare, iar funcția nu se va executa. Argumentele necesare sunt utile atunci când anumite date sunt esențiale pentru ca funcția să își îndeplinească sarcina.
- **Argumente opționale**: Argumentele opționale sunt parametri care au valori implicite atribuite în semnătura funcției. Dacă apelantul nu furnizează o valoare pentru aceste argumente, funcția va folosi valoarea implicită. Argumentele opționale permit dezvoltatorilor să facă o funcție mai flexibilă și să ofere comportamente implicite, permițând totuși personalizarea.
- **Liste de argumente cu lungime variabilă**: PHP acceptă, de asemenea, liste cu argumente cu lungime variabilă prin utilizarea `func_num_args()`, `func_get_arg()` și `func_get_args()`. Acestea permit dezvoltatorilor să creeze funcții care pot accepta un număr variabil de argumente. Listele de argumente cu lungime variabilă sunt utile atunci când numărul exact de intrări este necunoscut sau poate varia semnificativ.

Informațiile pot fi transmise funcțiilor prin lista de argumente, care este o listă de expresii delimitată prin virgulă. Argumentele sunt evaluate de la stânga la dreapta, înainte ca funcția să fie numită efectiv (evaluare *eager*).

Să ne uităm la un exemplu pentru a ilustra utilizarea argumentelor funcției PHP:

Exemplul #1 Utilizarea argumentelor funcției PHP

```
// Funcție cu argumente necesare
function addNumbers($a, $b) {
return $a + $b;
}

// Funcție cu argumente necesare opționale
function greetUser($name, $greeting = "Hello") {
return $greeting . ', ' . $name . '!';
}

// Funcție cu listă de argumente cu lungime variabilă
function calculateSum() {
$sum = 0;
$numArgs = func_num_args();

for ($i = 0; $i < $numArgs; $i++) {
$sum += func_get_arg($i);
}

return $sum;
}
```

În exemplul de mai sus, funcția `addNumbers` preia două argumente necesare și returnează suma acestora. Funcția `greetUser` preia un argument obligatoriu și un argument opțional cu o valoare implicită. Dacă apelantul oferă salutul, acesta va fi folosit; în caz contrar, „Hello” va fi folosit ca salut implicit. Funcția `calculateSum` demonstrează utilizarea listelor de argumente cu lungime variabilă. Aceasta calculează suma tuturor argumentelor transmise, indiferent de numărul de argumente.

PHP acceptă transmiterea argumentelor după valoare (implicita), trecerea prin referință și valorile argumentelor implicite. Listele de argumente cu lungime variabilă și argumentele denumite sunt, de asemenea, acceptate.

Exemple: (1)

Exemplul #2 Transmiterea de matrice la funcții

```
<?php
function takes_array($input)
{
echo "$input[0] + $input[1] = ", $input[0]+$input[1];
```

```
}
?>
```

Începând cu PHP 8.0.0, lista de argumente a funcției poate include o virgulă finală, care va fi ignorată. Acest lucru este util în special în cazurile în care lista de argumente este lungă sau conține nume lungi de variabile, ceea ce face convenabil să listați argumentele pe verticală.

Exemplul #3 Lista de argumente ale funcției cu virgulă finală

```
<?php
function takes_many_args(
$first_arg,
$second_arg,
$a_very_long_argument_name,
$arg_with_default = 5,
$again = 'un șir implicit', // Această virgulă finală nu a fost permisă
înainte de 8.0.0.
)
{
// ...
}
?>
```

Transmiterea argumentelor prin referință

În mod implicit, argumentele funcției sunt transmise după valoare (astfel încât, dacă valoarea argumentului din cadrul funcției este modificată, aceasta nu este modificată în afara funcției). Pentru a permite unei funcții să își modifice argumentele, acestea trebuie să fie transmise prin referință. (1)

Pentru a avea un argument pentru o funcție transmis întotdeauna prin referință, adăugați un ampersand (&) la numele argumentului în definiția funcției:

Exemplul #4 Transmiterea parametrilor funcției prin referință

```
<?php
function add_some_extra(&$string)
{
$string .= 'și ceva în plus.';
}
$str = 'Acesta este un șir, ';
add_some_extra($str);
echo $str; // ieșire 'Acesta este un șir, și ceva în plus.'
?>
```

Este o eroare să treci o valoare ca argument care ar trebui să fie transmisă prin referință.

Concluzie

Argumentele funcției sunt esențiale în programare, deoarece permit dezvoltatorilor să creeze cod versatil și reutilizabil. Acceptând diferite intrări, funcțiile pot îndeplini o gamă largă de

sarcini fără a fi nevoie să fie rescrise pentru fiecare caz specific. Acest lucru promovează eficiența codului și mentenabilitatea. În plus, argumentele funcției contribuie la lizibilitatea codului, deoarece dezvoltatorii pot înțelege scopul și cerințele unei funcții prin simpla examinare a semnăturii acesteia.

În concluzie, argumentele funcției PHP sunt un aspect fundamental al limbajului care permite dezvoltatorilor să creeze funcții dinamice și flexibile. Înțelegând diferitele tipuri de argumente și utilizarea acestora, dezvoltatorii pot scrie cod mai eficient, mai ușor de întreținut și mai reutilizabil. Pe măsură ce vă aprofundați în dezvoltarea PHP, stăpânirea artei argumentelor funcționale va deveni, fără îndoială, o abilitate crucială în construirea de aplicații web puternice și scalabile.

Referințe

(1) PHP Manual, licența CC BY 3.0. Traducere și adaptare: Nicolae Sfetcu